24, Avenue du Président Wilson
87700 Aixe sur Vienne

Cédric DESSET
DUT Informatique
Année 2006/2007

Intégration dans une équipe de développement afin de participer au développement d'un outil de gestion pour un client

Stage du 26 Mars 2007 au 02 Juin 2007

Tuteur de stage : Jimmy Jouannaud
Maître de stage : Thierry Crouzillat

Allée André AUROIS
87065 Limoges
Cedex

Remerciements :

Je tiens à remercier mon maitre de stage Thierry Crouzilla ainsi que Marc Cornuot pour l’aide qu’ils m’ont apporté lors de l’analyse et du développement du projet.

Je remercie aussi Sylvette Dache pour les informations nécessaires au développement du projet.

Je remercie Laurent Penot pour toutes les aides diverses qu’il a apporté durant la durée du stage.

Je remercie toutes les personnes de ProximIT pour le cadre de travail agréable.

Sommaire

Introduction 4

Présentation de l'organisme d'accueil 4

I. Analyse 4
1) Sujet du stage 4
a) Le but du projet 4
b) Analyse de l'existant 4
c) Cahier des charges 4

2) MERISE 4
a) Modèle Conceptuel de Données 4
b) Type de bases de données existantes 4
c) Les traitements 4

3) UML 4
a) Diagramme de use case 4
b) Diagramme de classes persistantes 4

II. Développement / Implémentation 4
1) Nouvelle Interface 4
a) Nouvelle fiche de non qualité 4
b) Justification des choix 4

2) Détails sur l'application 4
a) Fonctionnalités 4
b) Arborescence des écrans de l'application 4

3) Parties difficiles du code 4
a) Code en javascript 4
b) Requête SQL/PHP 4

III. Bilans 4
1) Bilan Personnel 4
a) Gestion du temps 4
b) Apport du projet 4

2) Expérience Professionnelle 4
a) Hiérarchie 4
b) Communication entre les services 4
c) Relation avec le client 4

Conclusion 4

Glossaire 4

Annexes 4

Introduction

ProximIT est une société de service née en mars 2005. Elle a une multitude de clients en France, en Europe et même dans le monde. Elle possède de nombreux domaines d'activités comme l'hébergement, le réseau, l'infogérance… Pour le moment, Proximit possède un siège social situé à Pompadour en Corrèze et une annexe à Aixe sur Vienne.

Parmi ses clients, la société *Notariat Services* héberge sur les serveurs de *ProximIT* leur intranet. Celui-ci permet de pouvoir gérer des anomalies ou des réclamations que les clients de *Notariat Services* peuvent émettre. C'est pourquoi je suis amené à réaliser la gestion des fiches de réclamations client ou de non qualité.

Dans une première partie, j'exposerai le travail réalisé concernant l'analyse du projet. Pour cela, j'énoncerai clairement le sujet posé tout en fournissant des explications.

Ensuite, j'expliquerai mes bases de travail pour réaliser l'analyse MERISE afin de réaliser la base de données. Et enfin, je ferai de même pour l'analyse UML.

Dans la seconde partie, j'expliquerai le travail effectué concernant le développement de l'application dans un langage de programmation. Je montrerai l'interface de l'application tout en justifiant mes choix. Ensuite, je montrerai les différentes fonctionnalités de l'application. Et enfin les divers points difficiles du code.

Dans la dernière partie, je ferai un bilan de ce stage. D'abord en faisant le point avec un bilan personnel. Puis en mettant l'accent sur l'expérience professionnelle acquise.

Présentation de l’organisme d’accueil

ProximIT est née de la volonté de plusieurs sociétés régionales (Limousin) d’externaliser leur service informatique. Plutôt que de laisser voir ces emplois disparaître de la région, un groupe d’ingénieurs et cadres issu de ces sociétés a décidé, en mars 2005, de créer *ProximIT* afin de répondre aux besoins économiques de ces acteurs locaux et ainsi, dans ce cadre, pouvoir offrir à d’autres sociétés leurs compétences et expériences.

ProximIT est une Société Anonyme à conseil d'administration qui compte environ trente employés. Elle dispose d’un capital social de 158 000,00 €. Son siège social principal se situe à Arnac Pompadour (Corrèze) mais il existe une annexe à Aixe sur Vienne (Haute-Vienne). Au cours de l’année 2005 (année de sa création), la société a réalisé un chiffre d’affaires de 930 000 €. A sa tête, on trouve un Président Directeur Général, Jérôme Barbier, un Directeur Général Délégué, François-Xavier Duny ainsi qu’un Directeur Commercial, Jean Lamige. Un organigramme de l’entreprise est disponible en annexe.

Cette société possède une activité principale qui est l’infogérance, proposant ainsi trois types de services :

- Architecture, gestion des matériels uniquement (serveurs, postes de travail, réseaux).

- Applicative, prise en charge du développement, de la maintenance et des éventuelles améliorations concernant les applications.

- Complète, prise en charge de l’ensemble des éléments constitutifs de l’environnement informatique (matériel et logiciel).

De cette activité principale en découlent diverses prestations telle que :

1 L’hébergement dans le domaine de l’Internet.

2 L’étude et le développement, en effet, *ProximIT* dispose d’une équipe complète, opérationnelle sur de nombreuses plateformes et sur de nombreux langages. Le pôle développement possède deux spécialisations fortes, les systèmes de gestion et les sites Internet ; permettant ainsi l’accès au système de gestion à partir de sites Internet.

3 La sécurité, d'un point de vue aussi bien logiciel que matériel.

4 L'audit et le conseil, apport de compétences de haut niveau dans des domaines précis (méthodologies et développement, maintenance, exploitation ou conduite de projet). Proposition de solutions ou bien d'un plan d'action à court et moyen terme.

5 Services D.S.I. (Direction de Systèmes d'Information), il s'agit de fournir une compétence de responsable Informatique à un organisme qui ne souhaite pas investir dans un poste à temps complet. Cette prestation comprend entre autre la gestion et le suivi des budgets, la rédaction du schéma directeur, les différents plans d'action ou encore le reporting à la direction, l'élaboration de tableaux de bord et le pilotage de prestataires internes ou externes.

6 L'entreprise propose également de s'occuper de l'aspect téléphonie pour ses clients. En effet, face au nombre grandissant d'opérateurs, d'offres et même de solutions en matière de communication vocale, *ProximIT* offre de s'occuper de tout pour mettre en place la téléphonie fixe et mobile.

De par la présence géographique de ses clients, la société *ProximIT* est implantée au niveau mondial. En effet, l'entreprise est en charge de la gestion de l'Informatique d'autres entreprises qui sont situées partout à travers le monde (Espagne, Brésil, Inde, Chine…). La société est en recherche continuelle de nouveaux clients, aussi bien en France qu'à l'étranger.

La société *ProximIT* s'appuie sur de nombreux partenariats avec de grands acteurs du marché, aussi bien au niveau mondial, national ou régional. Ces alliances s'inscrivent au cœur de la stratégie de l'entreprise, à travers une démarche de service client et une logique de création de valeur. Les principaux partenaires sont Orange Business Services (Partenaire expert), Le Studio Vert (possédant également des locaux à Aixe sur Vienne), Dell…

Les clients sont souvent des entreprises qui ont des besoins d'hébergement ou bien qui ne possèdent pas de service Informatique. Par exemple, le CRT du Limousin (Centre Régional du Tourisme) fait appel à *ProximIT* pour l'hébergement d'une application ainsi qu'une base de données et un serveur WEB. Le Conseil Général de la Haute-Vienne dispose également de deux serveurs (un serveur physique d'archives et un serveur virtuel pour la plateforme WEB). La société réalise l'infogérance du groupe Sicame, groupe international implanté dans 15 pays dont le secteur d'activité est le transport et la distribution d'électricité.

Poste de travail à Aixe-sur-Vienne - ProximIT

Le service dans lequel j'ai travaillé était le service des études et développements. Comme dit précédemment, cette branche de la société utilise de nombreux langages de programmation comme par exemple le java, php, html… et bien d'autres encore. Cette équipe comporte bien entendu un chef de projet, se nommant Thierry Crouzillat, avec qui j'ai travaillé en étroite collaboration, ainsi que Marc Cornuot qui était mon responsable du code. Il y a d'autres personnes qui font partie de cette équipe de développement mais je n'ai pas travaillé avec elles. De plus, cette équipe de développement a en particulier deux fortes spécialisations. La première est tout ce qui concerne les systèmes de gestion, et l'autre est le développement de site internet. L'intérêt de ces deux spécialisations est que l'on peut les fusionner afin d'obtenir des résultats simples à l'utilisation et très performants. En effet, la capacité de pouvoir gérer à partir d'Internet un système de gestion quelconque est une chose très efficace et avantageuse pour une entreprise qui utiliserait cette solution. De plus elle pourra gagner du temps si un problème survient car il n'y aura plus la nécessité de revenir sur un poste de travail de l'entreprise afin de régler le problème.

Présentation du thème de stage et des activités

L'énoncé de mon sujet est le suivant :

« Intégration dans une équipe de développement afin de participer au développement d'un outil de gestion pour un client. »

Le client pour qui je dois développer cet outil de gestion est le groupe *Notariat Services*. Situé à Arnac Pompadour, cette société conçoit et diffuse pour la profession notariale des journaux, des revues et des sites Internet immobiliers. De plus, Notariat Service travaille en partenariat avec *ProximIT* pour l'hébergement de leurs sites Internet, les développements informatiques, l'infogérance et la sécurité de leurs systèmes.

Bâtiments de Notariat Services

L'objectif de *Notariat Services* est de donner satisfaction à ses clients grâce à une amélioration continue de ses services. Ce groupe a donc certifié l'ensemble de leurs activités en ISO 9001 par l'AFAQ (Association Française d'Assurance Qualité) depuis 1998. Ce label qualité assure aux clients une organisation qui permet de prendre en compte leurs attentes, de les aider dans la réalisation de leur publicité, dans l'utilisation d'Internet et de leur logiciel de négociation.

Lorsqu'un client signale une erreur, *Notariat Services* intervient dans les 48 h pour compenser l'erreur, l'expliquer et s'excuser. Pour que chaque erreur soit traitée le plus rapidement possible, on l'étudie directement sur support informatique grâce aux Formulaires de Non Qualité (FNQ) disponible sur leur intranet. Suite à cette étude, il est

possible de déclancher un processus d'action améliorante qui sera lui aussi informatisé grâce aux FAC (Formulaire d'Action Corrective) disponible aussi sur l'intranet.

Mon travail est donc de réaliser ce FNQ. Ce formulaire existant déjà, je dois simplement mettre à jour celui-ci selon les demandes de certaines personnes dans l'entreprise. Pour cela, les personnes concernées m'ont donné des consignes pour améliorer ce FNQ ainsi que certains éléments importants qui doivent figurer. Par exemple, certaines rubriques changeront de nom, d'autres auront des questions en plus ou encore la possibilité de paramétrer certain composant du formulaire… Je précise que je n'utilise l'existant que pour réaliser une nouvelle analyse. Celle-ci étant effectuée à partir de l'ancien FNQ et en utilisant le cahier des charges qui a été réalisé par le client.

Afin de réaliser ce projet, mon maître de stage m'a demandé de réaliser deux types analyses au début. La première est une analyse MERISE. Celle-ci a été réalisée dans le but de créer la base de données qui permettra une historisation de chacun des formulaires. La seconde est une analyse UML. Celle-ci est une approche orientée objet et permettra de mettre en place la structure du travail à réaliser ainsi que les fonctionnalités de ce projet. Après avoir réalisé ces deux analyses, Thierry a organisé une réunion avec Marc Cornuot (personne de l'équipe de développement) et moi. Cela nous a permis de corriger mes analyses. Pour MERISE, il n'y avait pas grand-chose à dire car, c'était correct et cohérent dans l'ensemble sauf qu'il manquait deux ou trois petites choses à rajouter comme par exemple l'ajout d'attributs dans des entités ou encore mettre des associations manquantes entre certaines entités. En ce qui concerne l'UML, nous avons eu beaucoup de problèmes car pour une raison que nous ignorons, tous les fichiers de mon analyse sont devenus corrompus, que ce soit les fichiers récents ou les fichiers de sauvegardes effectuées. Aussi bien que sur mon poste de travail que sur celui de mon maître de stage. Donc à sa demande, nous avons mi de côté l'analyse UML car nous ne voulions pas perdre plus de temps afin de pouvoir nous consacrer au développement.

Pour ce qui concerne les moyens matériels, la société a mis à ma disposition un ordinateur de bureau avec comme système d'exploitation *Windows XP*. De plus, cet ordinateur était suffisamment puissant pour faire tourner des logiciels de développement qui sont parfois gourmand en ressources. Pour les logiciels utilisés, je ne pouvais utiliser que des logiciels libres ou ceux qui étaient déjà installés sur mon poste. Afin de réaliser l'analyse UML, mon maitre de stage m'a conseillé d'utiliser le logiciel *Poséidon*. Ce logiciel reste suffisamment instinctif pour réaliser les différents diagrammes. Cependant, ce logiciel comporte un défaut : il ne respecte pas la normalisation du diagramme de cas d'utilisation consistant à mettre les noms des cas d'utilisations en dessous de sa représentation schématique. Afin de réaliser l'analyse MERISE, j'ai pris l'initiative d'utiliser la version d'évaluation de *Power AMC*. Comme ce logiciel était une version d'évaluation,

il était limité à 15 jours de fonctionnement. Ce délai était suffisamment grand pour que je puisse faire le MCD ainsi que de générer la base de données. Pour la partie développement, j'ai utilisé le logiciel *Dreamweaver* qui était déjà installé sur mon poste de travail. Même si ce logiciel prend beaucoup de place et de ressource de la machine, il reste idéal pour le développement en html, php et javascript car il faut de la coloration syntaxique et il aide beaucoup le développeur car lorsque l'on appuie sur des touches clés, un panel de choix apparaît. De plus, il ferme automatiquement toutes balises que l'on a ouvert, se qui permet d'éviter de nombreuses erreurs dû aux balises qui n'ont pas été fermés correctement.

Mon maître de stage, Thierry Crouzillat, restait le plus possible à ma disposition en cas de problème. Je lui envoyais le soir par mail, le travail que j'effectuais afin qu'il puisse voir mon rythme de travail ainsi que le travail en lui-même. Lorsqu'il trouvait une erreur, il faisait en sorte de me mettre sur la piste et non de corriger l'erreur. Comme ça, cela me permettait de chercher et de comprendre l'erreur et part la suite de ne plus faire le même genre d'erreur. Même si il est chef de projet, il reste pédagogue et prend le temps d'expliquer clairement ses attentes. Pour ce qui concerne la partie développement, aussi bien Thierry que Marc attendent de voir la vitesse à laquelle je programme afin de voir si je ferai tout ou si Marc m'aidera un peu pour que l'application soit opérationnelle avant la fin du stage. Le développement s'est donc effectué dans les langages du web. Pour cela, j'ai dû installer sur la machine qui était à ma disposition un serveur apache / mysql. C'est un principe dans cette entreprise de ne mettre que sur les serveurs destinés à la mise en production que les projets finis et opérationnels.

Afin de réaliser ce projet, j'avais établi un planning prévisionnel qui est le suivant :

- 2 semaines pour les analyses MERISE et UML
- 6 semaines pour le développement
- 1 semaine pour faire des améliorations de certaines parties du code
- 1 semaine pour le manuel d'utilisateur et la mise en service

I. Analyse

1) Sujet du stage

a) Le but du projet

Le but de ce projet est en quelque sorte, d'améliorer le « service client ». Comme dit précédemment, *Notariat Services* tient énormément compte des critiques ou des remarques que peuvent faire n'importe quelle personne qui contacterait la société ou même un collaborateur en interne peut déclencher la création d'une fiche de non-qualité.

A partir du moment où la FNQ est créée (c'est-à-dire que les champs obligatoires du formulaire ont été renseignés), celle-ci va suivre une procédure (décrite en détail dans la partie 2.c.) qui permettra de corriger une ou des erreurs, de signaler d'éventuels problèmes. Par exemple une annonce sur leur site Internet « Immonot » (site immobilier notarial) peut contenir une erreur sur le nom de la personne à contacter.

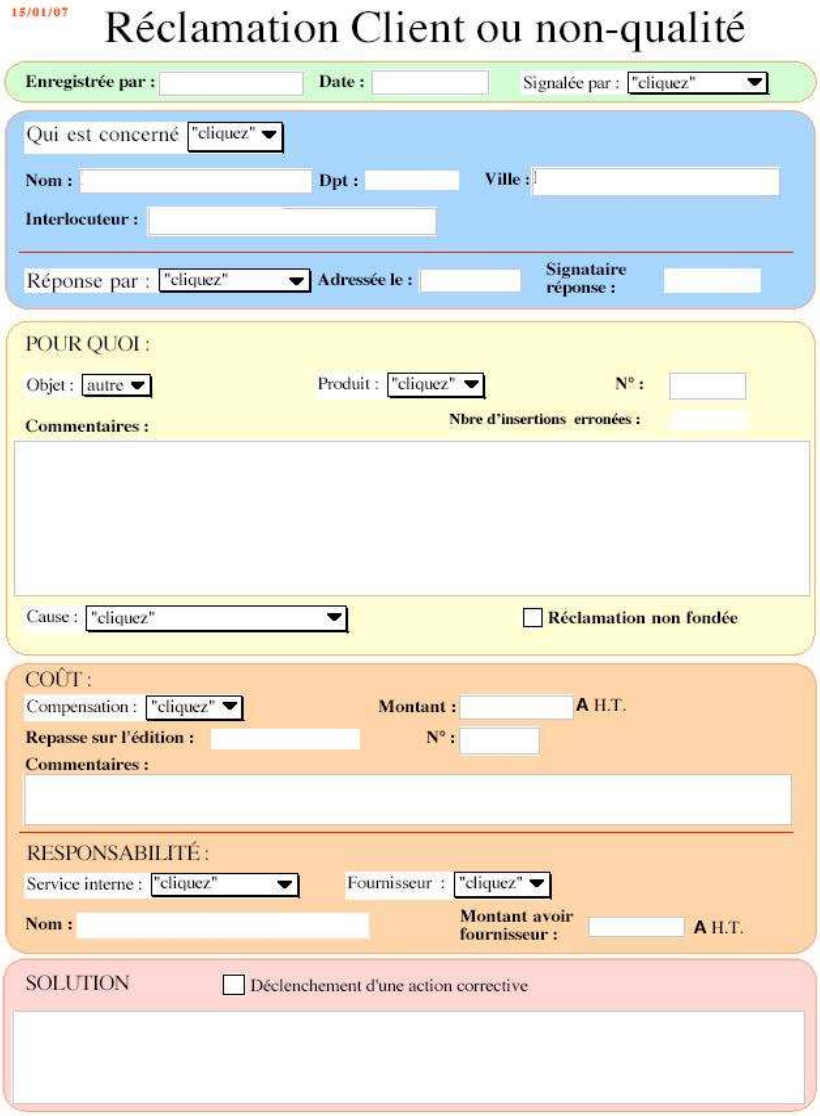
A la fin de cette procédure, une réponse est toujours envoyée à la personne ou au service interne de la société concerné afin qu'il soit au courant de la ou des solutions qui vont être mises en œuvre pour régler le problème.

b) Analyse de l'existant

Ce projet existant déjà, il a fallut en prendre connaissance et assimiler rapidement son fonctionnement afin de prendre conscience du projet global qu'il va falloir réaliser.

Pour ce, c'est le client même qui m'a expliqué le fonctionnement, la procédure que chaque fiche doit respecter…

Cette fiche est la fiche de non-qualité originale et qui est en fait, la base de tout mon travail.


15/01/07

Réclamation Client ou non-qualité

Enregistrée par : Date : Signalée par : "cliquez"

Qui est concerné "cliquez"

Nom : Dpt : Ville :

Interlocuteur :

Réponse par : "cliquez" Adressée le : Signataire réponse :

POUR QUOI :

Objet : autre Produit : "cliquez" N° :

Commentaires : Nbre d'insertions erronées :

Cause : "cliquez" Réclamation non fondée

COÛT :

Compensation : "cliquez" Montant : A H.T.

Repasse sur l'édition : N° :

Commentaires :

RESPONSABILITÉ :

Service interne : "cliquez" Fournisseur : "cliquez"

Nom : Montant avoir fournisseur : A H.T.

SOLUTION Déclenchement d'une action corrective

Fiche de non-qualité vierge déjà existante

Elle est simple dans l'ensemble. Il y a certaines zones comme par exemple « enregistrée par » et « date » qui sont renseignés automatiquement. D'autres zones comme « qui est concerné », « nom », « dpt » et « ville » sont des zones obligatoires. C'est-à-dire que l'enregistrement de la fiche ne pourra pas se faire tant que ces zones ne seront pas renseignées. De plus, les libellés sont suffisamment précis afin que quelque soit la personne qui va être chargée du traitement dans la société, elle puisse remplir la fiche de non-qualité sans avoir trop de questions à se poser.

Sans trop rentrer dans le détail, une personne quelconque peut remarquer une erreur sur un site Internet ou sur un journal d'informations de la société *Notariat Services*.

Cette même personne va signaler l'erreur par mail, courrier, téléphone ou autre à la société. A la réception de l'information par *Notariat Services*, la personne qui a reçu le message va déclencher la création d'une fiche de non qualité. Elle va donc saisir les différentes informations dans la fiche avec le contenu du message qu'elle a reçu. Puis elle va la transmettre à un collaborateur qui sera chargé de finir de compléter la fiche et de la valider.

c) Cahier des charges

Suite aux explications détaillées que le client m'a fournies, nous avons établi le cahier des charges.

Ce ne fût pas chose facile. En effet, il a fallut concilier les exigences du client et ce qui est vraiment réalisable. Durant cette réunion, nous avons d'abord commencé par modifier la fiche de non-qualité car celle-ci ne convenait plus au besoin de la société *Notariat Services*.

Ensuite, nous avons établi des contraintes comme par exemple imposer la saisie de certains champs et donc les rendre obligatoire. De même, d'autres champs ne doivent pas être modifiables par l'utilisateur. Je prends l'exemple du « login » utilisé pour se connecter à l'intranet de la société. Celui-ci apparaît sur la fiche de non qualité afin de savoir qui a créée cette fiche. Cela parait donc normal de figer ce champ afin qu'il n'y ait pas d'usurpation d'identité. De plus, il est tout aussi normal d'effectuer le même traitement sur la date de création de la fiche afin qu'aucune triche ne soit permise.

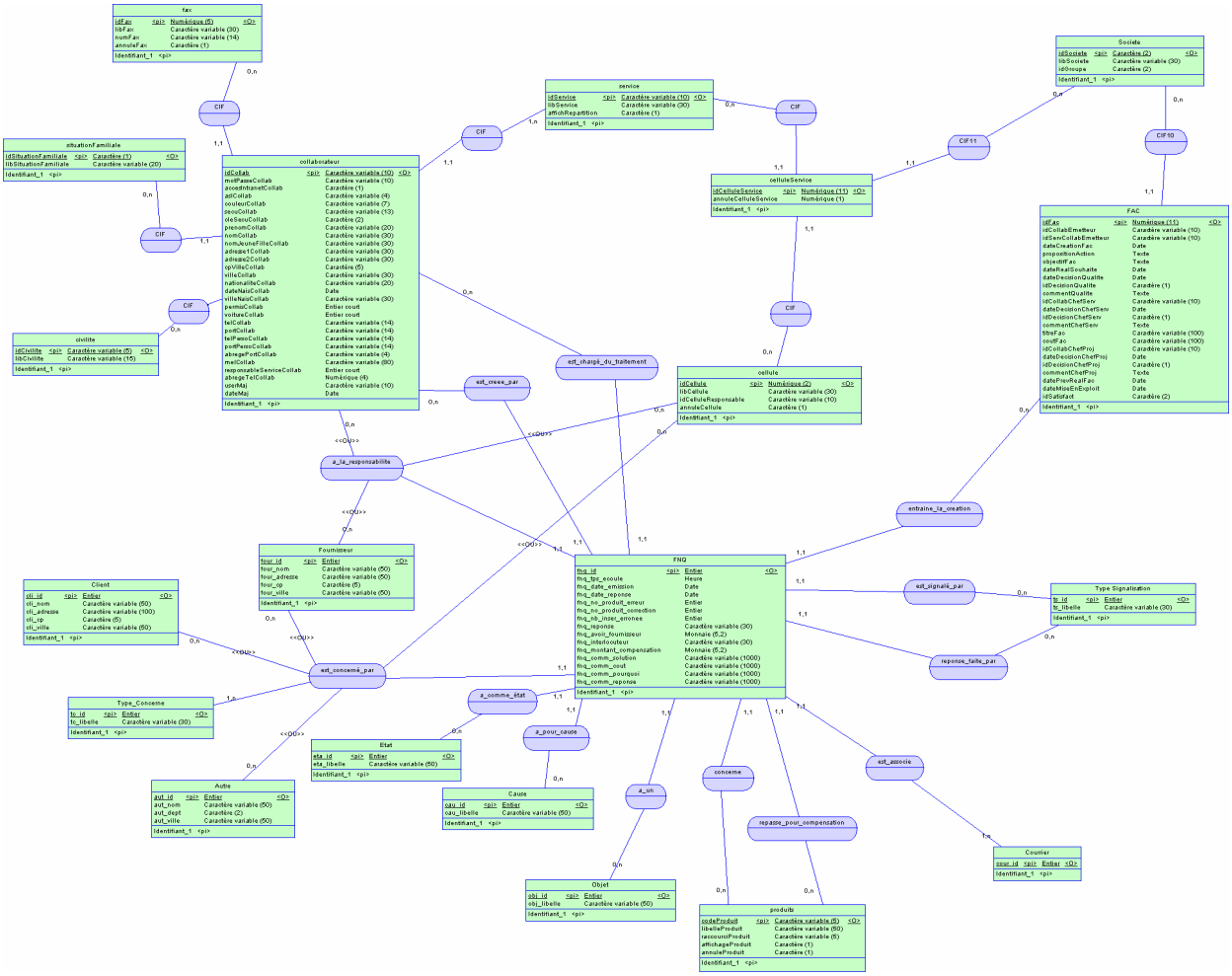
Pour ce qui concerne les autres éléments de la fiche de non qualité, j'avais beaucoup de liberté. Ce qui a été précisé lors de la première rencontre avec le client, est que l'application doit être opérationnelle même si dans l'ensemble ce n'est pas super joli. Un informaticien doit être polyvalent mais il y a des limites qu'il ne peut pas dépasser. Pour ça, il existe d'autres personnes qui ont les capacités qui manquent comme par exemple pour moi, un web designer aurait pu enjoliver mon application.

2) MERISE

a) Modèle Conceptuel de Données

Ce schéma d'analyse a donc était réalisé avec l'aide de la fiche de non qualité existante ainsi que du cahier des charges.

Il reste pour le moins simple dans l'ensemble malgré le grand nombre d'entités. Il a fallut prévoir l'enregistrement de tous les différents champs de la FNQ.



MCD global du projet

Dans ce MCD, l'entité « FNQ » regroupe beaucoup d'éléments de la fiche car si sur le point de vue d'un analyste il est possible de mettre des attributs dans d'autres entités. Mais il fallait tout de même se poser la question sur le point de vue efficacité. Je prends l'exemple de deux attributs qui sont dans l'entité « FNQ ». Les attributs sont : montant_comprensation et montant_avoir. Dans le MCD, ces deux attributs sont dans la même entité. Du point de vue d'un analyste, ajouter une entité « Coût » avec les attributs « cout_id » et « cout_valeur » que l'on relierai par deux associations « a pour montant » et « a pour avoir » avec des cardinalités à « 1,1…1,n ». Je n'ai pas retenu cette solution car il n'est pas nécessaire de créer une entité uniquement pour des coûts et qui ne sont pas forcément redondant.

Il y a aussi deux relations n-aires. Elles permettent de récupérer respectivement qui concerne la fiche et qui en est le responsable. L'autre point important de ces relations, est qu'il ne peut y avoir qu'un responsable et qu'une personne concernée. C'est pourquoi, des liens ont le stéréotype « OU » afin de respecter le cahier des charges.

Le MCD ci-dessus reprend en plus des entités que j'ai créées, celles déjà existantes comme par exemple les entités « Client », « Fournisseur », « Collaborateur »... Il a fallut reprendre exactement l'ensemble des propriétés afin de ne pas avoir de problème lors de la phase de développement.

Néanmoins, le schéma d'analyse précédent est un schéma global. Il permet de bien voir la totalité des entités car certaines comme « Client », « Fournisseur »… sont déjà utilisées pour d'autres applications. Il est donc normal que ces tables ne doivent pas être répliquées car des informations pourraient se perdre.

C'est pour cela qu'il a fallut que j'adapte le MCD précédent afin de pouvoir faire en sorte qu'il s'harmonise avec les bases de données existantes.

b) Type de bases de données existantes

Comme dit précédemment, le MCD reprend donc des entités appartenant à des bases de données différentes.

Il y a deux bases de données distinctes que j'ai dû utiliser. La première est une base « mysql » comme celle qui est à l'iut. Pour celle-ci, aucun problème n'a été rencontré pour accéder à son contenu car la base se situe sur un serveur « classique ». En revanche, la seconde base m'a posé plus de problème. En effet, celle-ci est une base dite « Claris » (Claris : équivalent de mysql pour les ordinateurs créés par *Apple*, c'est-à-dire les MAC). De plus celle-ci est située sur un serveur AS400.

Photo d’un AS400

Un AS400 est un mini-ordinateur conçu par IBM en février 1987. De plus, son architecture est composée d’éléments matériels et logiciels comportant une base de données et des éléments de sécurité très avancés. Sa force réside dans la modularité de ses éléments constitutifs lui conférant un haut degré d’adaptabilité et de sécurité.

Le système d’exploitation OS/400 est un système multiutilisateur, c’est-à-dire que plusieurs personnes peuvent travailler sur l’ordinateur simultanément, notamment grâce à un terminal relié à l'AS400.

En effet l'architecture AS/400 est une structure en couche séparant les éléments matériels et logiciels. Le système d'exploitation de l'AS/400 est appelé OS/400. Il repose sur une couche appelée MI (Machine Interface) chargée de fournir un ensemble de fonctions (API, application programming interface) que les applicatifs doivent utiliser afin de s'interfacer avec le matériel.

Contrairement à la plupart des systèmes d'exploitation, la gestion de la plupart des composants matériels (mémoire, entrées-sorties, gestion des processus, ...) est régie par une couche indépendante, appelée SLIC (System Licensed Internal Code), située sous la couche MI. Ainsi l'architecture AS400 assure une réelle indépendance entre le matériel, le système d'exploitation et les applications.

Représentation schématique des couches de l'AS400

Celui-ci comprend un écran et un clavier. Les écrans classiques de l'AS400 sont passifs, ils ne permettent donc pas une gestion de la souris ou de l'affichage de nombreuses couleurs. Pour ce qui est du client Notariat Services, les utilisateurs de l'AS400 utilisent un produit d'émulation pour macintosh (interAccess), qui simule un écran passif.

c) Les traitements

Toujours avec le cahier des charges et certaines données que le client m'a donné, il a fallut déterminer les différents traitements et la procédure que chaque fiche de non qualité devra suivre afin qu'elle soit prise en compte et non supprimé car la procédure n'aurait pas été respecté.

Le schéma suivant résume la procédure que la fiche de non qualité devra suivre. Par ailleurs, quelques précisions s'imposent pour la totale compréhension du schéma :

- Chaque action qui est représenté par des rectangles (sauf pour l'envoi ou la transmission de la fiche de non qualité) ne doit pas forcement être exécuté directement après l'action précédente. Plusieurs jours peuvent se passer entre les actions.

- Création d'une FAC (fiche d'action corrective).

  Une FAC est une fiche qui est aussi remplie par l'intranet de la société afin de pouvoir déclencher une action pour améliorer ou corriger un problème dans l'entreprise.

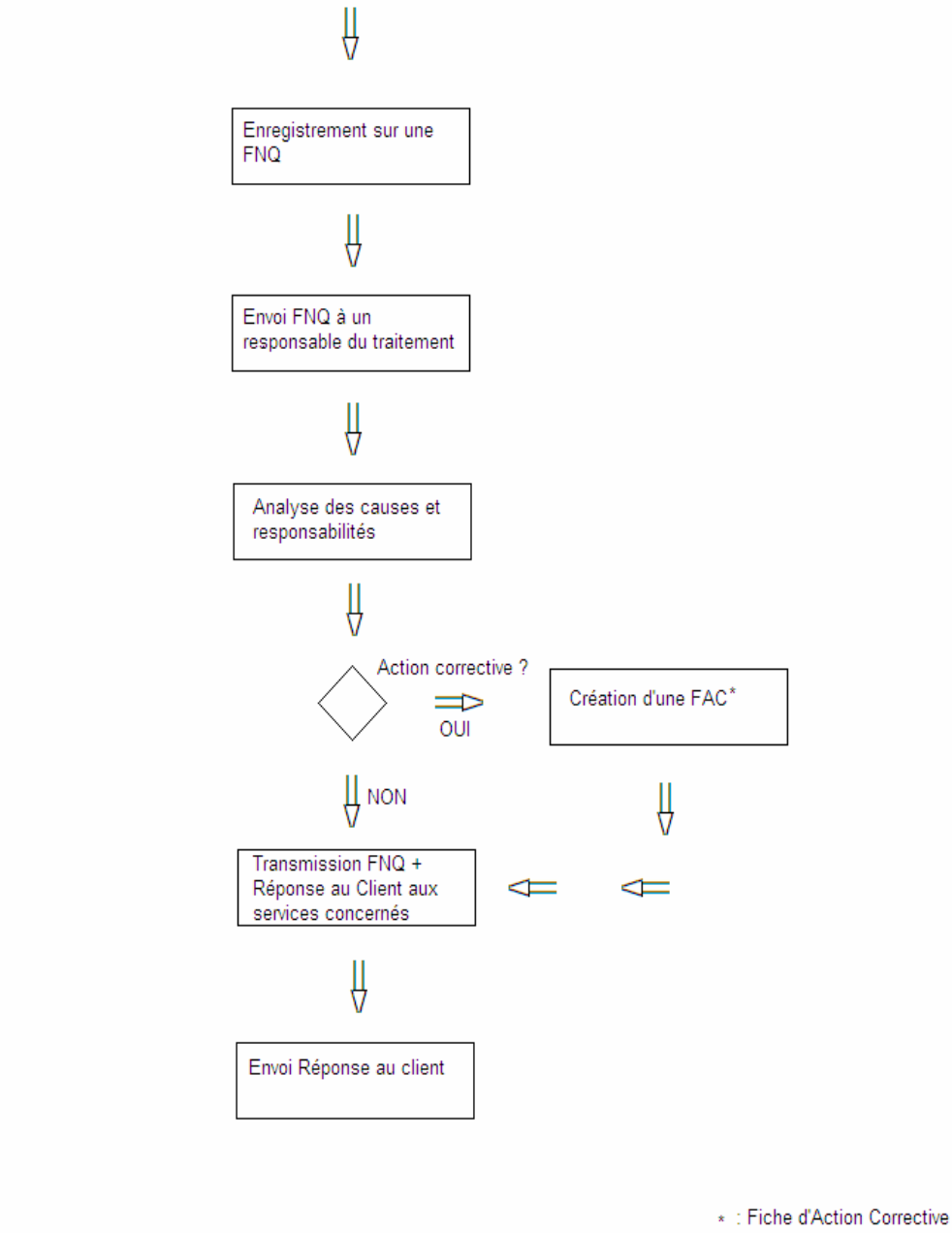

Réclamation CLIENT ou dysfonctionnement INTERNE
Enregistrement sur une FNQ
Envoi FNQ à un responsable du traitement
Analyse des causes et responsabilités
Action corrective ?
OUI
Création d'une FAC*
NON
Transmission FNQ + Réponse au Client aux services concernés
Envoi Réponse au client
* : Fiche d'Action Corrective

Schéma de la procédure que doit suivre une fiche de non qualité

3) UML

a) diagramme de use case

Le but de l'analyse UML était de la génération des classes afin de pouvoir réaliser le développement.

A partir de la procédure qu'une FNQ doit respecter, j'ai pu établir le diagramme de cas d'utilisation ci-dessous.

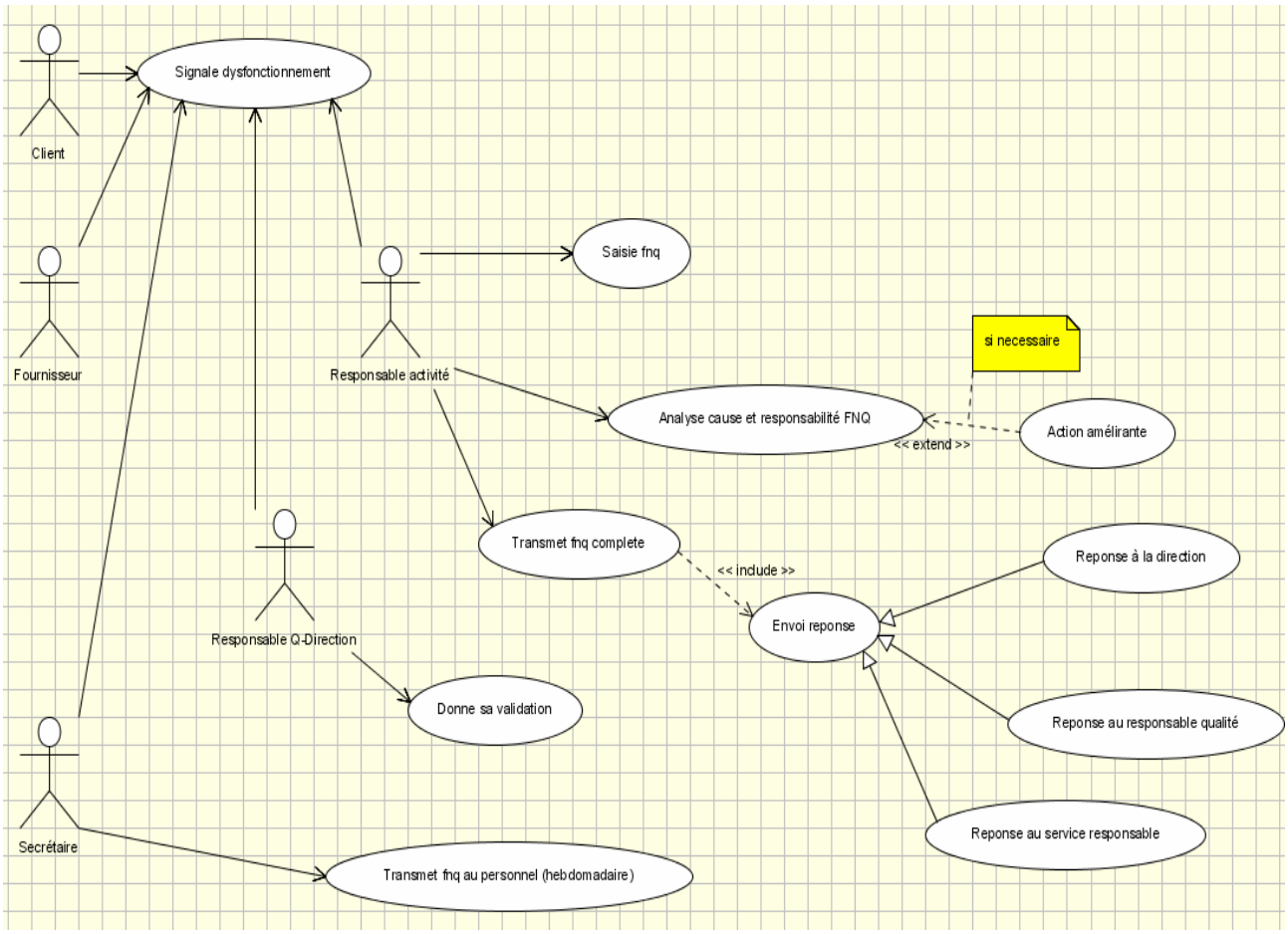


Diagramme de Use Case fait avec Poséidon

On est parti du principe que tout le monde peut signaler un dysfonctionnement, ou erreur ou simplement faire un constat pour tout ce qui a un rapport avec *Notariat*

Services. Par exemple, un client peut s'apercevoir qu'il manque des informations sur un site web. Ensuite, après ce signalement, un responsable va saisir une fiche de non qualité avec l'aide des informations du signalement de la personne.

Une fois saisie, cette fiche va être transmise à un responsable qui va s'occuper d'analyser les causes de cette fiche de non qualité ainsi que de la répartition des responsabilités si responsabilité il y a. Ensuite, des courriels seront envoyés automatiquement aux services concernés. Pour finir, seule la direction peut valider ou non la fiche de non qualité ce qui permet de lancer un processus pour subvenir à la demande initiale.

b) Diagramme de classes persistantes

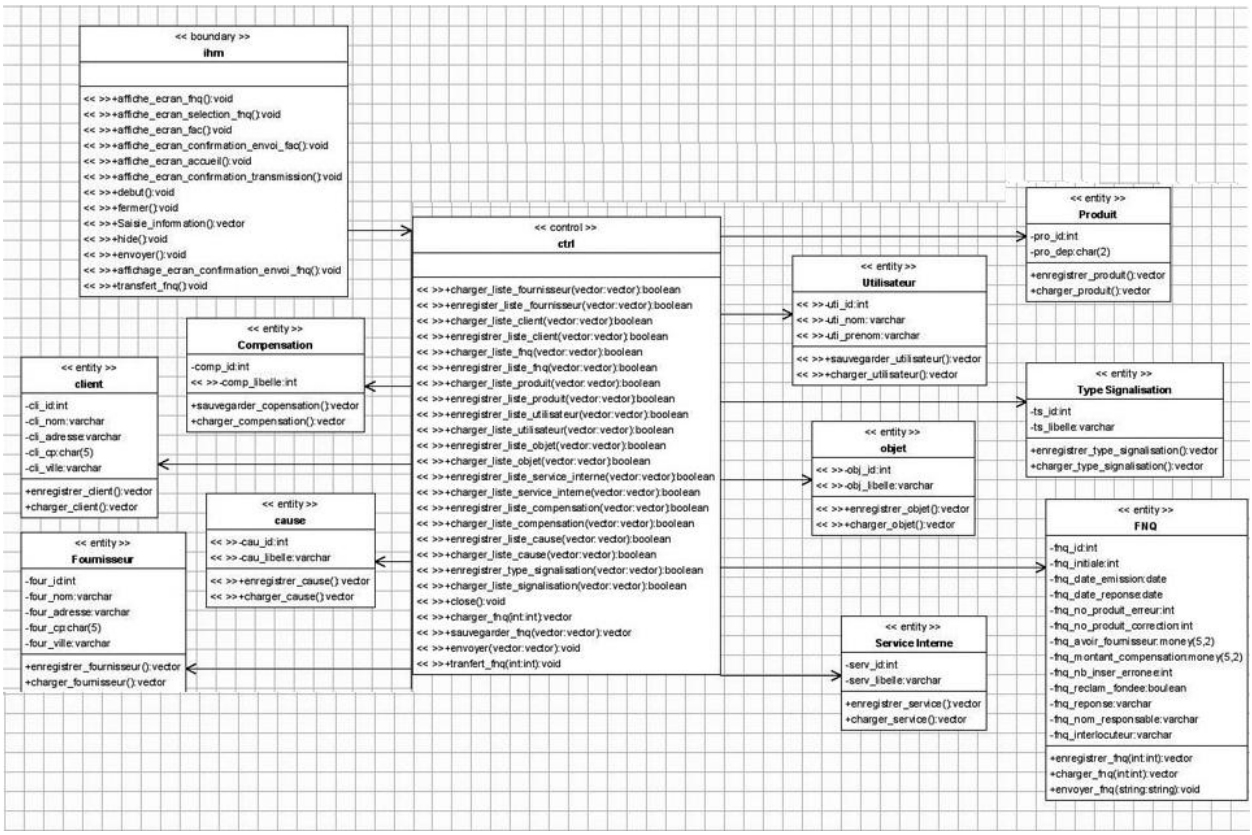


Diagramme de classes persistantes fait avec Poséidon

Le diagramme de classe ci-dessus est donc une « photo » de la future base de données. On y retrouve bien une « entity » FNQ destinée à conserver les différentes informations de la fiche de non qualité. Il y a bien sur d'autres « entity » correspondant aux tables déjà existantes sur les serveurs de *Notariat Services*.

Chaque fois que l'on souhaitera récupérer des informations sur une fiche, un client ou autre, on récupérera les informations en passant par le contrôleur.

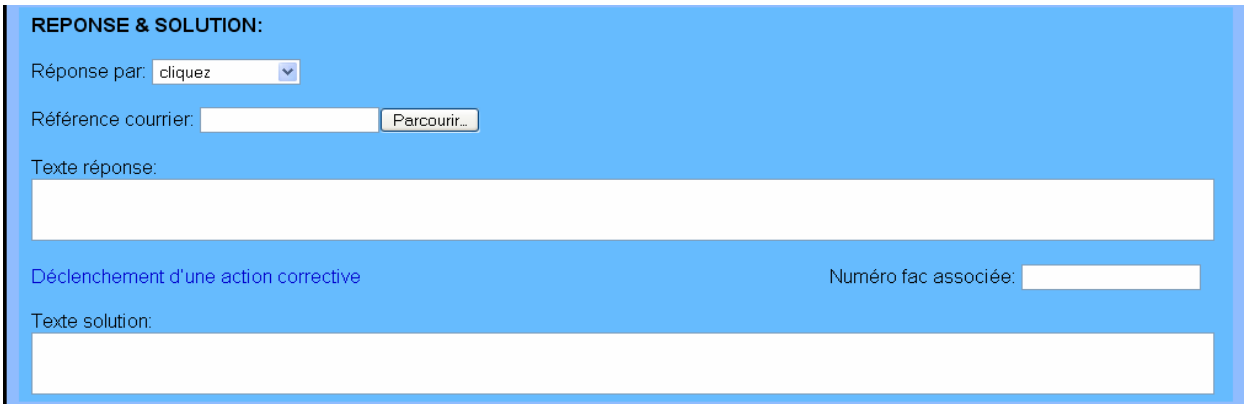
Suite à un problème avec *Windows*, je n'ai pas pu générer les classes car au moment où j'ai de nouveau ouvert mes fichiers de l'analyse UML, ceux-ci sont devenus corrompus. Même mon maitre de stage n'a pas réussi à les ouvrir. A sa demande, nous avons mis de côté cette analyse car elle n'a pas pu aboutir avec la génération des classes. Cependant grâce à l'analyse MERISE, le projet pourra être mené à terme.

A la fin de la partie analyse, il a donc fallut se lancer dans la partie développement. Celui-ci a été réalisé dans les langages du Web (html, php, javascript).

II. Développement / Implémentation

1) Nouvelle Interface

a) Nouvelle fiche de non qualité



Nouvelle fiche de non qualité vierge

b) Justification des choix

En m'inspirant de l'ancienne fiche de non qualité et en m'appuyant sur la demande du client, j'ai réalisé la nouvelle FNQ. Pour ne pas bousculer le futur utilisateur, j'ai conservé la position de beaucoup d'éléments de la fiche afin que l'utilisateur puisse directement prendre en main l'application. Le fait de déplacer des éléments importants peut faire tromper l'utilisateur au début de la prise en main. Comme les fiches de non qualité sont très importantes pour le client, le choix de conserver la même place pour les éléments de la fiche, est le plus judicieux.

Le choix des couleurs est aussi justifiable. L'application qui a été développé, va être mise sur l'intranet de la société. Celle-ci a déjà un type de navigation, ici la particularité est que la page est séparée en deux. Le haut de la page est réservé à un menu dynamique (qui s'affiche ou non suivant si la souris passe dessus ou pas). Tandis que le reste de la page est consacrée à l'affichage des informations telles que la fiche de non qualité ou même encore les fiches d'actions correctives. L'intranet a aussi ses propres couleurs. J'ai donc repris les mêmes couleurs qu'une autre application (les fiches d'actions correctives) afin de rester dans le même ton que l'intranet et les applications déjà existantes.

La fiche de non qualité ci-dessous est une fiche complète vierge. En réalité, l'application ne permet pas de saisir une fiche en une seule fois. Elle a été coupée en deux. La première partie s'arrête jusqu'au choix du responsable du traitement et c'est ensuite que vient la seconde partie. Ce découpage existe car tout le monde dans la société accédant à l'intranet peut créer une fiche de non qualité, mais elle n'est pas forcément capable de la remplir d'un seul coup. C'est pourquoi on désigne un responsable de traitement qui pourra terminé de remplir la fiche et la valider. A cause de ce découpage, il a fallut rajouter un attribut « état » qui permet de savoir à quel stade est une fiche.

Bien entendu, la quasi-totalité des champs où l'on saisie du texte sont contrôlés afin que l'on ne puisse pas, par exemple, saisir du texte dans un champ où l'on doit saisir un nombre pour que lors de l'enregistrement en base, il n'y ait aucun problème. Il y a aussi un contrôle qui vérifie si les champs obligatoires sont bien renseignés.

2) Détails sur l'application

a) Fonctionnalités

La détermination des différentes fonctionnalités que l'application devra être capable de faire n'est pas forcément le plus facile. Il est difficile de prévoir dès le début toutes les fonctionnalités. Pour ce qui me concerne, j'avais oublié une fonctionnalité permettant d'effectuer une recherche suivant différents critères comme par exemple le nom du responsable de traitement, la date de création ou encore le numéro de la fiche.

Liste des fonctionnalités :

- création d'une FNQ : partie où l'on ne saisie que la première partie de la FNQ

- validation d'une FNQ : on saisie que la seconde partie et on peut enregistrer temporairement une fiche ou la valider

- consultation d'une FNQ : permet d'afficher le contenu d'une fiche

- recherche : permet de rechercher des FNQ selon divers critères

- modification d'une FNQ : permet de modifier une fiche uniquement par son créateur ou une personne du service « Qualité » à condition que l'état d'une FNQ soit « Créé » (autrement dit, la première partie de la FNQ)

- suppression d'une FNQ : permet de supprimer une fiche uniquement par son créateur à condition que l'état de la fiche soit « Créé »

- statistique : permet de faire afficher différents types de statistiques

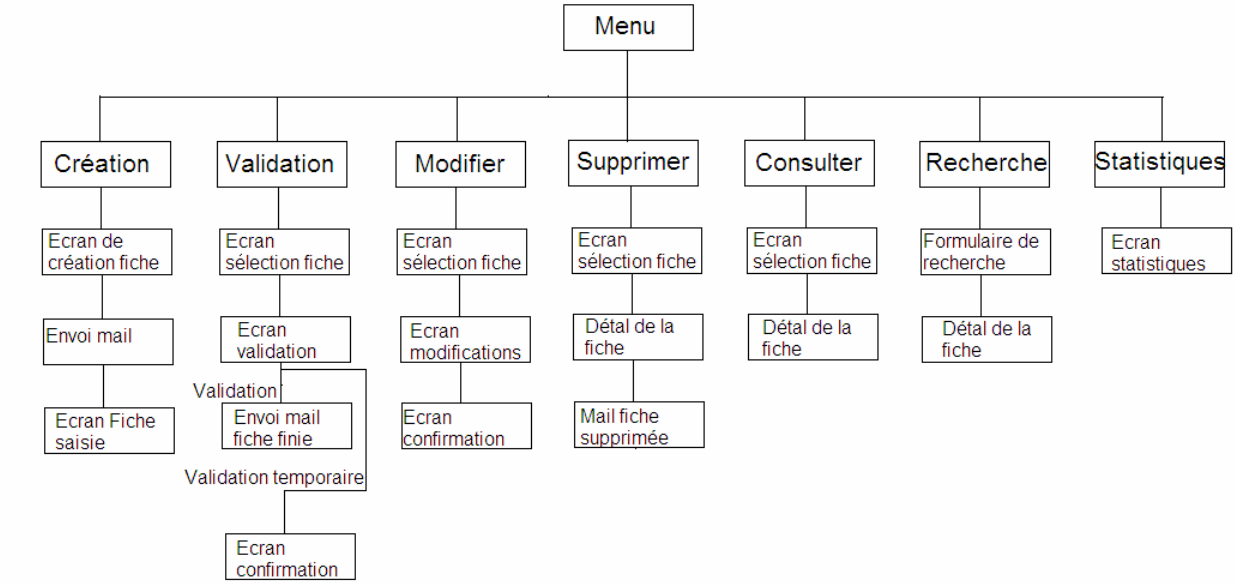
- mail : permet d'envoyer un mail à un collaborateur l'avertissant qu'il est responsable d'une fiche, qu'il n'est plus responsable d'une fiche…

- liste : permet d'établir la liste des FNQ dont est responsable le collaborateur qui s'est connecté à l'intranet, ou la liste des FNQ que le collaborateur peut supprimer, ou la liste des FNQ que le collaborateur peut modifier, ou la liste des FNQ que le collaborateur peut valider

b) Arborescence des écrans de l'application

Afin de bien se faire une idée de l'ensemble de l'application, j'ai choisi de présenter les différents écrans sous forme d'arbre. Je pense que cela permet de mieux voir le fonctionnement de l'application.

L'arborescence ci-dessous ne représente que les écrans principaux. Pour des soucis de lisibilité, je n'ai pas rajouté tous les pop-up affichant des messages de confirmation ou d'erreur.



Arborescence globale de l'application

3) Parties difficiles du code

a) Code en javascript

Lors du développement, il y a certaines parties du code qui été plus dures car en cours, je n'ai pas vu précisément les différentes syntaxes du langage javascript.

D'abord, le nombre de caractères que l'on peut saisir dans la fiche doit être limités car quelque soit le type d'un attribut dans la base de données, il a toujours une taille maximale. C'est pourquoi il a fallut créer la fonction suivante :

```
//fonction qui limite en taille le nb de caractère saisie
function countChars(area, maxChars)
{
```

```
		if(area.value.length > maxChars) //si nombre de caractère max atteins
		 {
			alert("Nombre de caractères maximum atteint"); //message d'erreur
			area.value = area.value.substr(0, maxChars); //suppression du dernier caractère
		}

	}
```

Lorsque l'on saisie quelque chose dans un champ, chaque fois que l'on ajoute ou supprimer un caractère, on appelle la fonction précédente. Lorsque le nombre maximal de caractères est atteint, il faut bien penser, à partir du moment où le nombre maximal de caractères est dépassé, à supprimer les caractères en trop sinon cette fonction ne sert pas.

Il y a aussi une autre fonction importante. Celle-ci permet de vérifier si ce que l'on saisie est bien de type entier et non réel car il existe une fonction isNaN() qui permet de vérifier si la saisie correspond à du texte ou des nombres entiers ou réels.

```
	//fonction vérifiant si un champ saisi est de type entier
	function verifTypeEntier(contenu,champ)
	{
		var valide = true;
		if(isNaN(contenu)) //le contenu est du texte
		{

			window.document.forms['fnq'].elements[champ].className='invalid';
			return false;

		}
		else // le contenu est un nombre réel ou entier
		{
			//on parcourt le contenu pour voir si il y a le caractère '.'
```

```
                for(var i=0;i<=contenu.length;i++)
                {
                        if(contenu.substr(i,1) == '.')
                        {
                                valide = false;

                        }
                }

                if(valide)
                {

window.document.forms['fnq'].elements[champ].className='valider';
                        return true;
                }
                else
                {

window.document.forms['fnq'].elements[champ].className='invalid';
                        return false;
                }
        }
}
```

Le principe est le suivant. D'abord, il faut vérifier que le contenu ne soit pas des lettres.

Ensuite il faut parcourir chaque caractère pour voir si il y a un le caractère '.' qui signifierai que le nombre est de type réel. Si ce n'est pas le cas, on peut valider le formulaire.

b) Requête SQL/PHP

Les requêtes qui permettent d'insérer ou de modifier une fiche de non qualité dans la base de données. Comme l'entité « FNQ » comporte beaucoup d'attribut, les requêtes sont donc assez conséquentes.

Exemple de requête pour modifier une FNQ en base :

```
$req="update fnq set cau_id=".$_POST['cause'].",
fnq_comm_cause='".$_POST['commentaire_cause']."',
fnq_avoir_fournisseur='".$_POST['montant_avoir']."',
fnq_montant_compensation='".$_POST['montant_compensation']."',
fnq_tps_ecoule='".$_POST['heure'].":".$_POST['minute']."',
fnq_no_produit_correction='".$_POST['num_correction']."',
pro_codeproduit='".$_POST['no_produit_correction']."',
fnq_comm_cout='".$_POST['commentaire_cout']."',
 typ_ts_id=".$_POST['rep_par'].",
".$tmp."
 fnq_comm_reponse='".$_POST['commentaire_reponse']."',
 idfac='".$_POST['fac']."',
fnq_comm_solution='".$_POST['commentaire_solution']."',
 eta_id=".$temp.$date." where fnq_id=".$_GET['id'].";";
```

Il existe aussi une autre difficulté dans la réalisation. Lors de la création d'une fiche de non qualité, il doit y avoir deux insertions dans deux tables différentes. La difficulté résidait dans le moyen de récupérer l'identifiant de la première insertion dans la table « FNQ » car elle a un identifiant en « auto-incremente ». C'est-à-dire que l'on n'a pas besoin de gérer ce numéro et qu'il est incrémenté automatiquement à chaque insertion.

L'astuce que j'ai trouvée est la suivante : il suffit de faire une requête permettant de récupérer l'id de la FNQ que l'on vient de créée en recherchant par l'identifiant du créateur tout en triant les identifiants des fiches par ordre décroissant.

```
//récupération de l'id de la dernière FNQ enregistré
$req3="SELECT fnq_id FROM fnq where idcollab='".$_POST['enregistre_par']."' order by fnq_id desc";
```

Ensuite pour la seconde insertion, il suffit de prendre l'identifiant que l'on récupère avec la requête précédente.

III. Bilans

1) Bilan Personnel

a) Gestion du temps

Le fait de travailler dans une entreprise et non dans le cadre scolaire, rajoute une plus grande pression. Dans ce cadre, il n'est plus question de points en moins sur une note par jour de retard. Mais c'est d'entretenir une bonne relation entre la société est le client.

Pour ce projet ci, qui est le plus conséquent réalisé jusqu'à ce jour, il était important de bien gérer mon temps pour mener le projet à bien dans les délais du stage.

D'abord, la partie consacrée à l'analyse aurait dû être plus courte qu'elle ne l'a été en réalité. Comme dit dans la partie concernant l'analyse UML, j'ai eu quelques problèmes à propos des fichiers qui étaient corrompus. Ce problème étant indépendant de ma volonté, m'a tout de même fait perdre du temps qui aurait pu être consacré au développement. Comme se sont des problèmes qui arrivent, il faut apprendre à les gérer pour essayer de terminer le projet dans les délais ou alors avec le plus petit retard possible.

Ensuite, pour la partie concernant le développement de l'application qui est beaucoup plus longue que l'autre partie, il fallait essayer de déterminer le temps que j'allais mettre pour développer chaque une des fonctionnalités. De plus, au fur et à mesure que le projet avançait, il arrivait que le client décide de changer ou rajouter des éléments. Cela à donc contribué à allonger le temps consacré au développement de l'application.

Au final, je peux dire que la bonne gestion du temps lors d'un projet est primordiale. Elle permet en outre de pouvoir donner au client une date de fin, ainsi qu'une vision globale des coûts par tâches et par ressources grâce à la gestion de projet.

b) Apport du projet

Même si ce projet a été réalisé dans les langages « html » et « php » que nous voyons à l'iut, cela ne m'a pas empêché d'approfondir mes connaissances et de découvrir de nouvelles syntaxes. En plus de la découverte de fonctionnalités de ces langages, j'ai pu apprendre concrètement le langage « javascript ». Ce langage ressemble un peu au langage « java » que nous avons vu en début d'année. Par contre, il est beaucoup plus difficile de bien le comprendre. Il a donc fallut que je m'adapte rapidement.

Ce projet m'a aussi apporté en plus d'apprendre à mieux gérer mon temps et l'assimilation d'un nouveau langage, c'est de croire en ses propres capacités. En effet, au début du projet je ne savais pas si je serai capable de le mener à terme. Puis au fur et à mesure que les jours passaient, je voyais que les briques du projet se montaient une à une afin de monter un mur complet. Le secret réside dans le fait que je suis resté prudent d'un bout à l'autre du projet.

De plus, ce projet est le premier réalisé en autonomie. Pendant mes trois années passées à l'iut, nous avons toujours réalisé les travaux pratiques en binôme. Dès le début, nous avons pris l'habitude de travailler en groupe ou en binôme mais jamais tout seul. Lors de ce projet, je n'ai pas pu déléguer de tâches.

Même si en réalité je n'étais pas seul, Thierry Crouzillat ainsi que Marc Cornuot étaient toujours disponible en cas de problèmes. Cependant, sur le point de vue travail sur le projet, j'ai tout effectué d'un bout à l'autre.

2) Expérience Professionnelle

a) Hiérarchie

Depuis mon arrivé dans l'entreprise, j'ai découvert un vrai lien de subordination. En effet, chaque fois que j'avais besoins d'informations diverses à propos du développement, des procédures ou autre à demander à l'entreprise cliente, je devais passer absolument par un de mes supérieurs direct (soit Thierry Crouzillat ou Marc Cornuot). En même temps, c'est une chose qui est tout à fait normale car je ne suis qu'un stagiaire dans cette entreprise. Je n'ai pas à prendre des initiatives pour contacter le client ou autre.

Même si ces deux personnes étaient mes supérieurs directs, ils n'ont jamais abusé de leur autorité. A chaque fois qu'il fallait opérer un changement sur quoi que se soit, ils demandaient toujours si cela me gênait et si ce changement était possible à réaliser avec mes connaissances.

De plus, même si ils étaient souvent occupés à faire diverses tâches, ils n'ont jamais refusé de m'apporter leur aide. Pour cela, ils font preuve d'une grande pédagogie et de patience.

b) Communication entre les services

La communication entre les différents services au sein d'une société est très importante. Comme chaque personne dans l'entreprise a une fonction précise, il est nécessaire de faire circuler les informations afin de les transmettre au bonne personne. Par exemple, la secrétaire peut recevoir un message pour une personne dans l'entreprise, il est donc nécessaire de transmettre ce message dans les plus brefs délais à la personne adéquate.

J’ai pu remarquer que beaucoup d’informations circulaient par mail. Chose non surprenante pour une entreprise travaillant dans le domaine de l’informatique, mais le fait est qu’il y a une volonté de vouloir garder une trace écrite. En plus de conserver les informations qui circulent, les mails sont un moyen de communication rapide et efficace. En plus, il permet d’être moins protocolaire et plus relâche. Elle méthode permet aussi de transmettre des informations importantes comme par exemple des mots de passes ou autres.

Cela n’empêche pas non plus la communication orale mais elle ne sert que pour des renseignements brefs comme par exemple demander l’adresse d’un serveur... Cela permet aussi d’avoir une réponse en direct et plus rapidement que par mail.

c) Relation avec le client

L’entretient d’une bonne relation avec le client est aussi une chose importante. Il faut établir un bon contact qui permettra de mener le projet à terme. C’est avec lui qu’il faut régler les détails qui sont resté flou dans le cahier des charges même si le client l’a signé.

Il arrive que certains détails puissent également lui échapper et c’est en conservant une bonne relation que les détails peuvent être réglés rapidement.

Pour ce qui me concerne, le client (*Notariat Services*) était représenté par Sylvette Dache. C’est donc ensemble que nous avons établi dans une bonne ambiance le cahier des charges. Lorsque celui-ci fût fini de rédiger, il lui a été envoyé. Elle a pris le soin de le lire et de corriger certaines choses qui étaient flou au moment de la rédaction puis ensuite, elle l’a approuvé.

Même lorsque le cahier des charges fût signé, cela ne m’empêchait pas de lui demander des informations supplémentaires pour bien comprendre certains éléments de la procédure.

Conclusion

En résumé, le travail effectué durant ces neuf semaines n'était pas sans difficultés. Il a d'abord fallu rencontrer le client afin de prendre connaissance de ses besoins et de la future application à réaliser. La première partie de mon travail consistait à bien comprendre l'existant et de faire les analyses pour pouvoir développer l'application par la suite. La seconde partie était le développement de l'application dans les langages du web. Le plus difficile dans cette partie était d'assimiler un langage (JavaScript) que nous avons juste survolé durant les cours. Les écarts entre le planning prévisionnel peuvent se justifier de la façon suivante. Lorsque j'avais besoin d'informations nécessaires pour continuer le projet, je demandais donc à mon maitre de stage ou Marc Cornuot. Mais ils n'avaient pas forcément le temps de me répondre immédiatement car ils avaient déjà un travail plus urgent que le mien mais ils ont toujours essayé de me répondre le plus tôt possible. Cependant, durant le laps de temps où j'attendais des informations, je m'avançais dans le code en développant une nouvelle fonctionnalité ou alors en améliorant des parties de code.

Au final, l'application développée pour le client *Notariat Services*, devrait être mise en service sur les serveurs dans les semaines qui arrivent. Le code contient suffisamment de commentaires afin qu'il puisse être reprit par quelqu'un d'autre. Il y a aussi la possibilité d'ajouter des fonctionnalités sans avoir à modifier tout le code.

Pour conclure, ce stage a eu pour but d'acquérir une première expérience professionnelle dans le domaine de l'informatique. Il nous a permis de voir le fonctionnement de différents services dans l'entreprise et de nous faire une idée sur celui-ci. D'ailleurs, j'ai été surpris de voir le fonctionnement de cette entreprise. Avant de visiter celle-ci, je m'attendais à voir les informaticiens dans des salles différentes. C'était une idée fausse. Ici, ils sont tous dans une ou deux salles et l'ambiance est très conviviale car tout le monde peut participer à chacune des discutions ce qui permet de travailler dans un cadre agréable.

Glossaire

AFAQ: Association Française d'Assurance Qualité

Apache: Un serveur Apache est un interpréteur de page web

AS400: Un AS400 est un mini-ordinateur conçu par IBM en février 1987.

Claris: Logiciel équivalent à Mysql ou PhpMyAdmin permettant d'administrer des bases de données pour les ordinateurs MAC

Entity: Stéréotype visant donner un "type" à une classe

Dreamweaver: Logiciel utilisé pour le développement web

FAC: Fiche d'Action Corrective

FNQ: Fiche de Non Qualité

Html: HyperText MarkupLangage utilisé pour créer des sites Internet

Javascript: Langage utilisé pour rendre l'Html dynamique

MCD: Modèle Conceptuel de Données

MERISE: Méthode d'analyse par les données

Mysql: Logiciel permettant d'administrer des bases de données

Php: Langage de programmation permettant de gérer des bases de données

Poséidon: Logiciel utilisé pour une analyse UML

Power AMC: Logiciel utilisé pour une analyse MERISE

UML: Unified Modeling Language. Méthode d'analyse avec une approche objet.

Annexes

Sommaire :

Organigramme ProximIT III

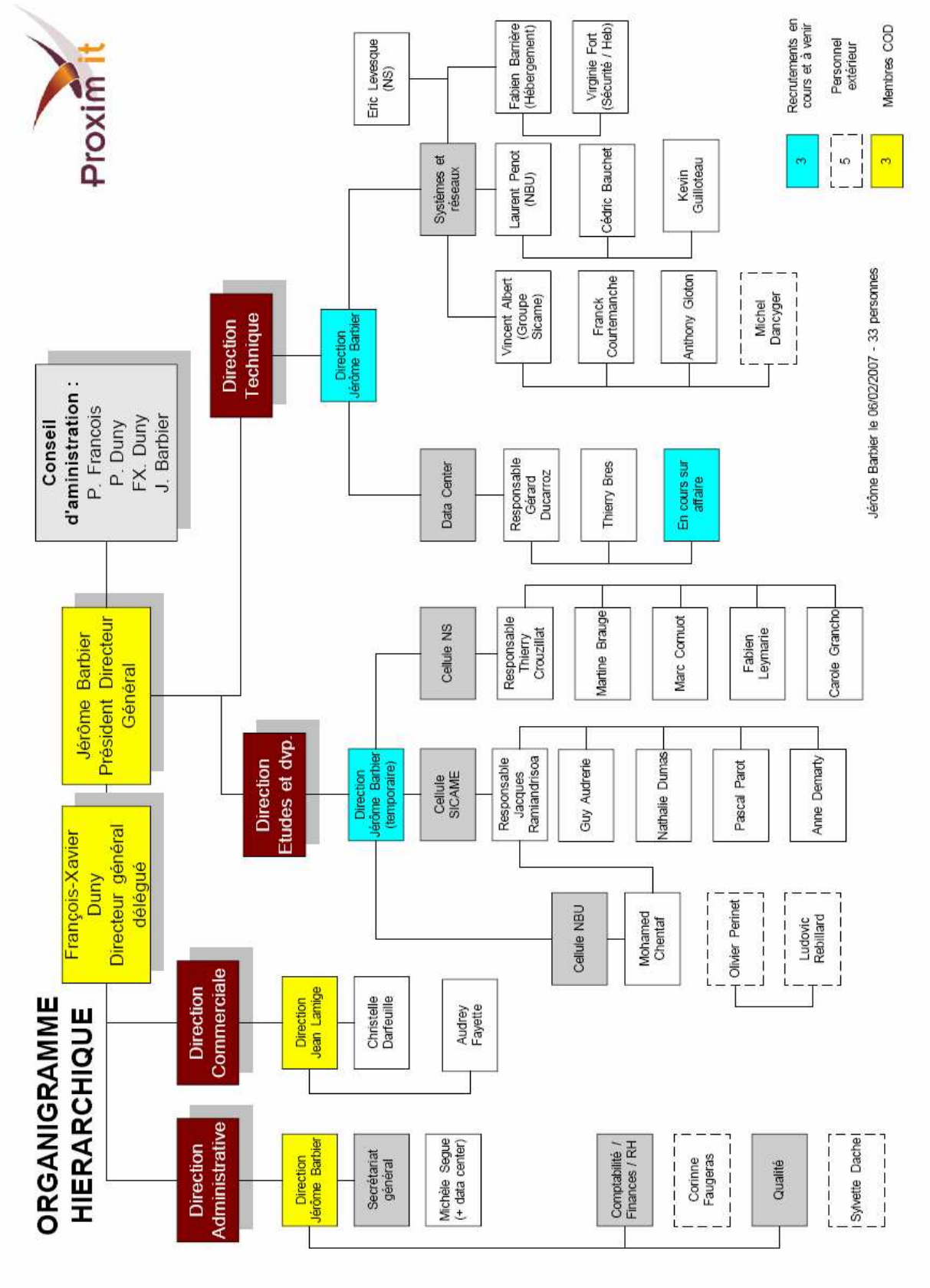
Réseau de ProximIT en France IV

Réseau ProximIT européen V

Réseau ProximIT mondial VI

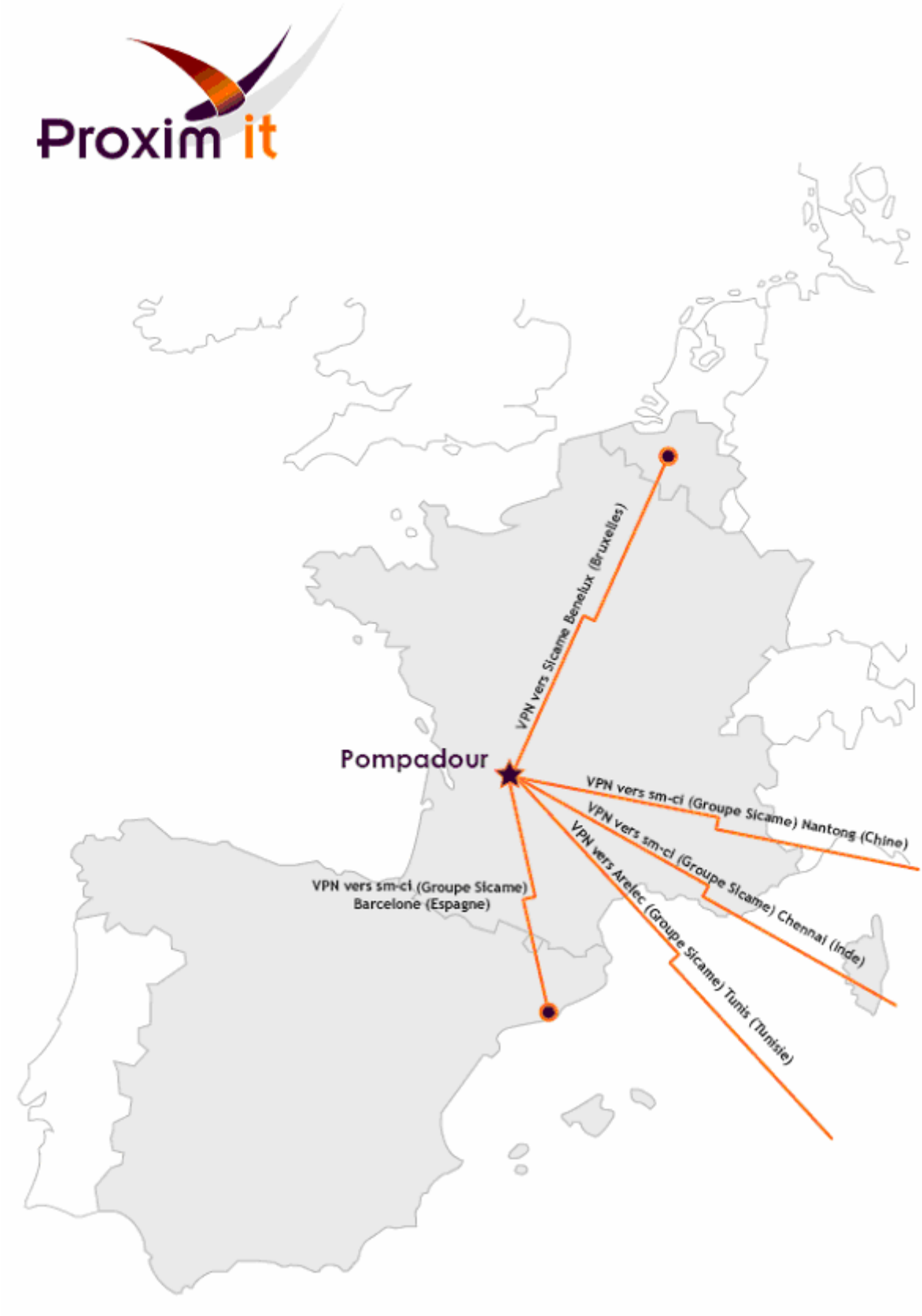
Planning réalisé VII

Organigramme ProximIT :

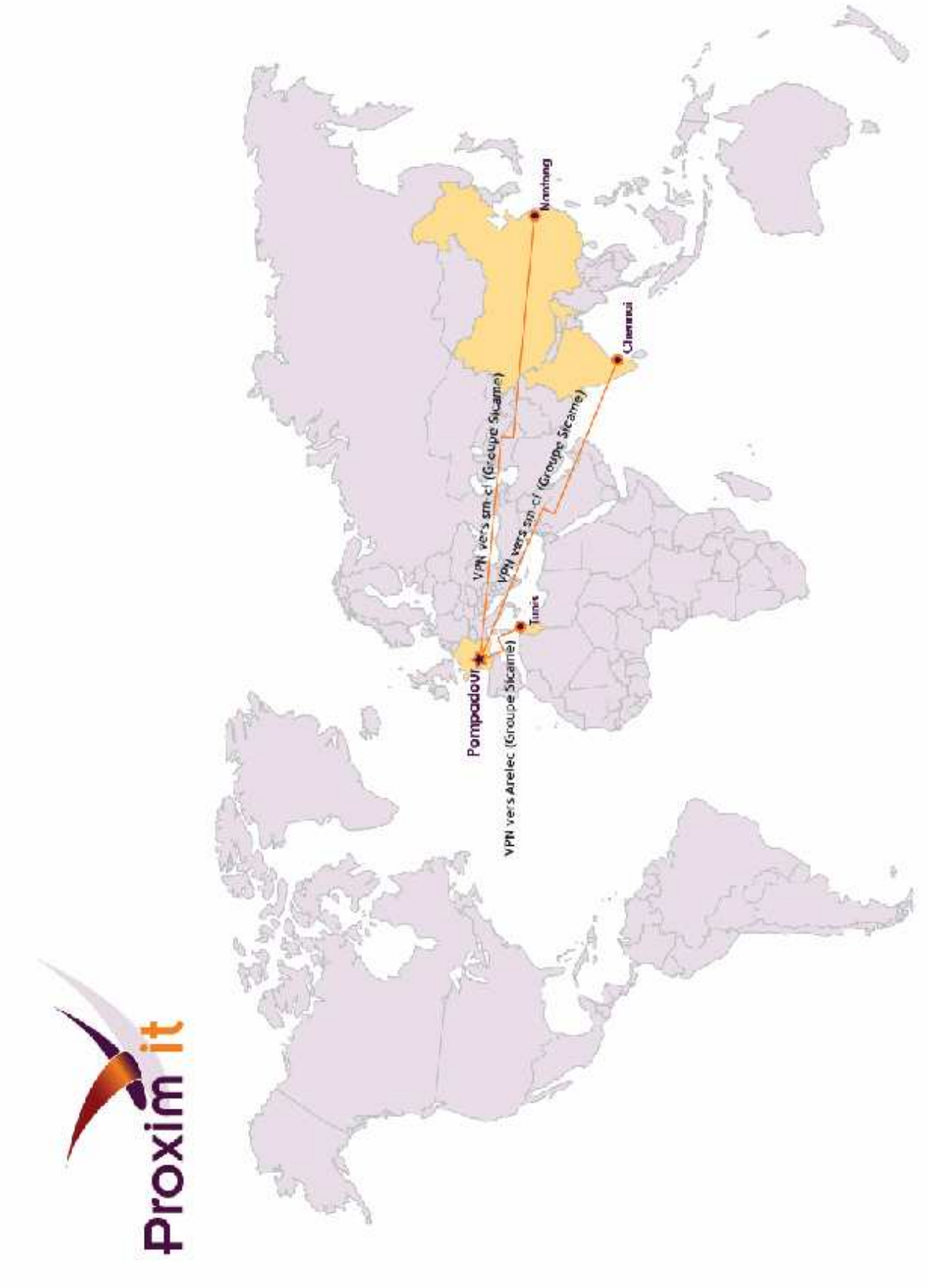


Réseau de ProximIT en France :

Réseau ProximIT européen :

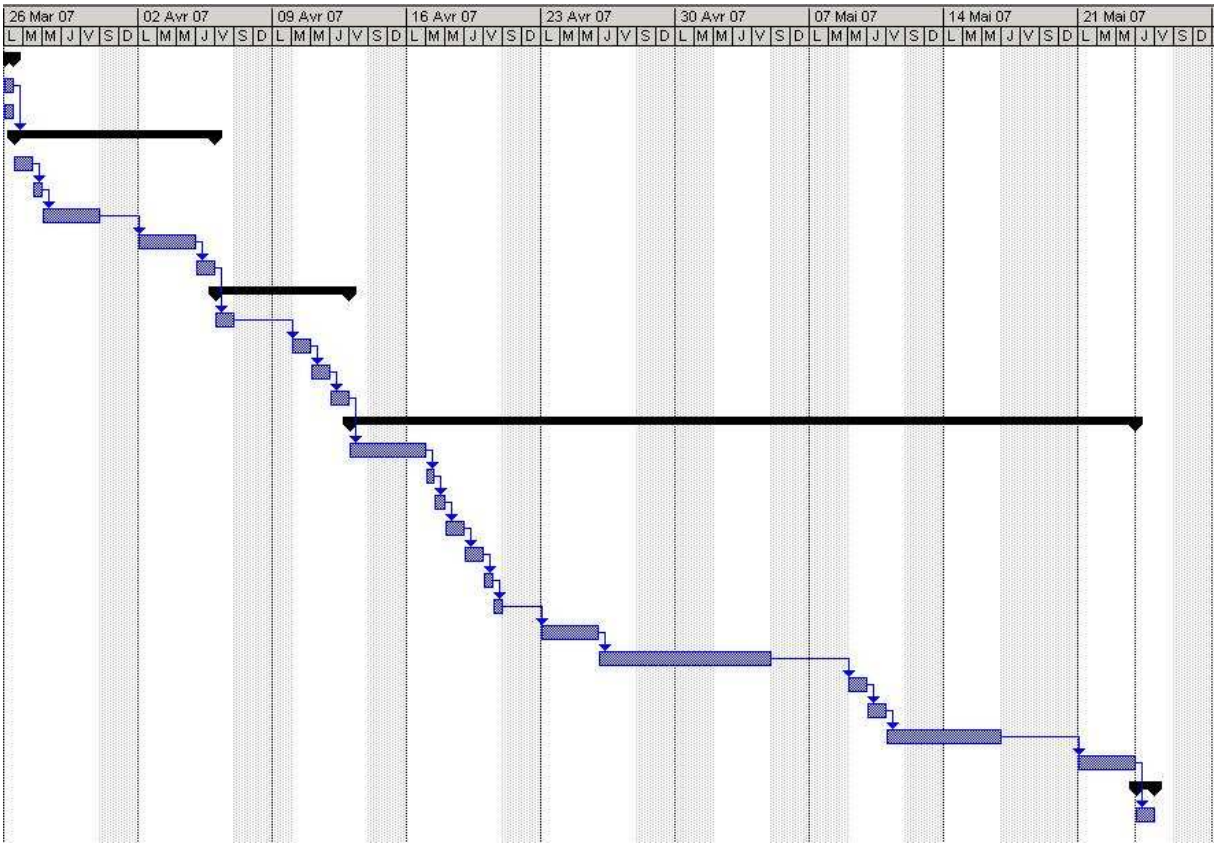

Proxim it
Pompadour
VPN vers Sicame Benelux (Bruxelles)
VPN vers sm-ci (Groupe Sicame) Nantong (Chine)
VPN vers sm-ci (Groupe Sicame) Chennai (Inde)
VPN vers Arelec (Groupe Sicame) Tunis (Tunisie)
VPN vers sm-ci (Groupe Sicame)
Barcelone (Espagne)

Réseau ProximIT mondial :



Planning réalisé :

		Nom de la tâche	Durée	Début	Fin	Prédécesseurs
1		**Début du projet**	**0,5 jour**	**Lun 26/03/07**	**Lun 26/03/07**	
2		Prise de connaissance du projet	4 hr	Lun 26/03/07	Lun 26/03/07	
3		cahier des charges	4 hr	Lun 26/03/07	Lun 26/03/07	
4		**Analyse MERISE**	**8,5 jours**	**Lun 26/03/07**	**Jeu 05/04/07**	**2**
5		Modèle Conceptuel de Données	1 jour	Lun 26/03/07	Mar 27/03/07	
6		Correction par maitre de stage	0,5 jour	Mar 27/03/07	Mar 27/03/07	5
7		Modification MCD	3 jours	Mer 28/03/07	Ven 30/03/07	6
8		Harmonisation avec cahier des c	3 jours	Lun 02/04/07	Mer 04/04/07	7
9		Génération de la base de donnée	1 jour	Jeu 05/04/07	Jeu 05/04/07	8
10		**Analyse UML**	**4 jours**	**Ven 06/04/07**	**Jeu 12/04/07**	
11		Diagramme de Use Case	1 jour	Ven 06/04/07	Ven 06/04/07	9
12		Diagramme d'activités	1 jour	Mar 10/04/07	Mar 10/04/07	11
13		Diagramme de séquence	1 jour	Mer 11/04/07	Mer 11/04/07	12
14		Diagramme de classe	1 jour	Jeu 12/04/07	Jeu 12/04/07	13
15		**Développement**	**23 jours**	**Ven 13/04/07**	**Mer 23/05/07**	
16		Squelette FNQ	2 jours	Ven 13/04/07	Lun 16/04/07	14
17		Lister FNQ	3 hr	Mar 17/04/07	Mar 17/04/07	16
18		Afficher FNQ	5 hr	Mar 17/04/07	Mar 17/04/07	17
19		Créer FNQ	1 jour	Mer 18/04/07	Mer 18/04/07	18
20		Valider FNQ	1 jour	Jeu 19/04/07	Jeu 19/04/07	19
21		Modifier FNQ	4 hr	Ven 20/04/07	Ven 20/04/07	20
22		Supprimer FNQ	4 hr	Ven 20/04/07	Ven 20/04/07	21
23		Recherche FNQ	3 jours	Lun 23/04/07	Mer 25/04/07	22
24		Ajout code javascript	5 jours	Jeu 26/04/07	Ven 04/05/07	23
25		Début manuel d'utilisateur	1 jour	Mer 09/05/07	Mer 09/05/07	24
26		Fonctions mail	1 jour	Jeu 10/05/07	Jeu 10/05/07	25
27		Récupération données sur AS40	4 jours	Ven 11/05/07	Mer 16/05/07	26
28		Statistique	3 jours	Lun 21/05/07	Mer 23/05/07	27
29		**Test**	**1 jour**	**Jeu 24/05/07**	**Jeu 24/05/07**	
30		test de l'application	1 jour	Jeu 24/05/07	Jeu 24/05/07	28


26 Mar 07
02 Avr 07
09 Avr 07
16 Avr 07
23 Avr 07
30 Avr 07
07 Mai 07
14 Mai 07
21 Mai 07
L M M J V S D